International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 12 Issue 10 ǁ October 2024 ǁ PP. 09-18

A Framework of Ubiquitous Bidirectional Translation for Language Learners

FATOKE, Tunde Joshua[1*], BOYINBODE, Olutayo Kehinde[1], ADETUNMBI, Adebayo Olusola[2], AKINWONMI, Akintoba Emmanuel[2]

[*1]*Department of Information Technology, Federal University of Technology, Akure, Ondo State, Nigeria*
[2]*Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria*
Corresponding Author: Fatoke Tunde Joshua tjfatoke@futa.edu.ng

Abstract
The trend in language translation systems that support language learners is increasing, particularly for low-resourced languages like Yoruba which is one of the languages speaking parts in Nigeria. The research points to a framework for ubiquitous bidirectional translation system in which emphasis is on Yoruba and English languages. It combines real-time translation capabilities, seamless mobility, and ubiquitous operations. It allows the users to translate text between Yoruba and English and vice versa on ubiquitous devices. In which transformer model with attention mechanism, mobile technology, and internet services are used. This provides a dynamic and user-friendly interface for language translation and learning activities. The translation model is design using neural machine translation (NMT) model – transformer which contains encoder and decoder phases with attention mechanism. There is significant improvement over rule-based and statistical models for low-resource languages. Through training the system which makes use of the bilingual corpus of Yoruba and English, the model learns to understand the different complex linguistic patterns, including tone, syntax, semantics and morphology, these are very crucial in Yoruba language. Also, the research makes use of external resources such as English – Yoruba dictionaries, cultural books, religious books and sentence context meaning to enhance the accuracy. Yoruba language is a tonal language where meaning can change with the same word based on pitch variations. The work supports bidirectional translation, emphasize on both English speakers learning Yoruba and Yoruba speakers learning English. It is designed to be ubiquitous, making learners to access translation services across various devices: smart watches or bracelets, mobile phones, tablets, and laptops. This makes users to engage in language translation and learning activities anywhere and anytime, in alignment with the principles of ubiquitous. The results shows that ubiquitous bidirectional translation for language learners majorly improves language communication and retention among language learners, making it an effective tool for language education and preservation of the languages.

Keywords: *Ubiquitous, Yoruba-English translation, Transformer model, Attention mechanism, Bidirectional translation, Low-resource languages, Natural Language Processing (NLP).*

Date of Submission: 01-10-2024 Date of acceptance: 11-10-2024

I. INTRODUCTION

Ubiquitous translation and learning aspect are emphasized in the work by making the translation system available across smart devices, mobile, tablet and laptops. Ubiquitous translation with learning environments provides learners with access to translating resources anytime, anywhere and on any smart device. This increases the opportunities for continuous engagement and immersion in language learning [6]. This system's accessibility aligns with the needs of both Yoruba and English learners, who may not have consistent access to formal language training classes or resources. Yoruba is a low-resource language primarily spoken in the western region of Nigeria, West Africa and it is not widely recognized on a global assessment. [5], it is a tonal language, it presents a distinct challenge in maintaining meaning across translations. Tone is used in Yoruba to differentiate words meaning especially words that have the same spelling or structure [3]. For instance, the word [k[can mean "husband ([k[)," "dagger (-k=)," "hoe ([k-)," or "vehicle ([k=)" depending on the tone applied and diacritic on the letters of the word. Transformer model, with attention mechanisms is capable of learning these tonal distinctions and ensuring that the context is preserved in the translation [8]. Attention process lets the model to focus on main segment of the sentence during translation, helping to handle the linguistic complexity of Yoruba. The Transformer model is now the state-of-the-art architecture in machine translation since its introduction, surpassing earlier models like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks in both accuracy and efficiency [31]. RNN model processes sequences of words in one after

the order while the Transformer architecture processes sequences of words together. It uses self-attention mechanisms that allow it to get relationship between or among words in a sentence irrespective of their positions. This ability to model long-range dependencies is particularly beneficial for Yoruba, which requires precise attention to both word order and tonal variations to maintain the meaning. By applying self-attention, the Transformer can better understand the context, tone and diacritics that are critical in translating Yoruba [31].

Furthermore, this research serves as an invaluable resource for both local and foreign speakers of Yoruba and English, fostering cross-linguistic understanding and language preservation. One of the main benefits of the Transformer model is its scalability and robustness in handling large datasets, which is good for modeling a low-resource language like Yoruba [25]. Additionally, the inclusion of positional encoding in the Transformer architecture helps it to recognize the order of words, which is vital for Yoruba, sentence meaning can change significantly based on word order and tone [8]. The work also contributes to language learning by providing an effective tool that addresses the unique challenges of translating between a high-resource language like English and a low-resource language like Yoruba.

II. RELATED WORKS

[12] makes use of ubiquitous applications and Statistical Machine Translation (SMT) system for English-to-Sanskrit translation. The research utilized Eclipse, an editor and emulator, along with a cross-compilation toolkit, in combination with Java for system development. By leveraging on Android's touch screen, sliding keyboard, and integration with Camera, GPS, compass, and accelerometer, the system discarded unnecessary desktop functionalities to create a streamlined mobile application. The focus of the study was to enhance the translation table and preprocessing Sanskrit text, but it did not include considerations for input handling, and the model remained unimplemented. The research contributed to translation efforts but lacked a complete deployment. In [13], the researchers further expanded on their work in ubiquitous language learning and translation using SMT. The study aimed to enhance translation quality through the statistical machine decoder, optimizing translation tables and text preprocessing. This framework integrated a client/server configuration, enabling the entire translation service to operate on mobile devices. By utilizing the mobile phone's capabilities, the system could support a ubiquitous learning environment. However, limitations in network technology at the time prevented large-scale implementation, highlighting the challenges of deploying such systems within expansive networks. [26] introduced the concept of a ubiquitous mobile real-time visual translator for the Bahasa language, combining augmented reality (AR) with mobile technology to support educational purposes. Visual Cognitive Simulative Software Development Life Cycle Methodology for Augmented Reality (V-CSSDLC-AR), aimed to translate Bahasa words into English in a real time which allows language learning in any location. It was only simulated and had not been fully implemented.

[4] focused on web-based English to Yorùbá machine translation system to improve on earlier Yorùbá language translation models. A rule-based approach, utilizing Context-Free Grammar (CFG), was employed to create a web-enabled platform for translating simple sentences. However, the system's vocabulary was limited, translating only a few words, which constrained its functionality and highlighted the need for a more comprehensive lexicon for effective translation. [11] pointed out Yorùbá machine translation by addressing tone changes in monosyllabic verbs. This is a very important part of Yorùbá grammar. Using a rule-based system developed with Python and the Natural Language Toolkit (NLTK), the system handled tone variations during translation from English to Yorùbá. The software was designed using Unified Modeling Language (UML) and could parse sentences effectively. However, it did not address serial verbs, an important feature of Yorùbá syntax, leaving room for further development in the context of machine translation for tonal languages. [6] emphasized on enhancing language learning through a mobile translation application that used Optical Character Recognition (OCR). The application was designed for Malay language translation, It also, allowed non-native speakers to understand Malay by converting text into English or Arabic. It used statistical machine translation and optimized distributed speech recognition with noise suppression for better performance in noisy environments. However, it supported only two languages (English and Arabic), and each language had to be configured before use, which limited its flexibility and scalability. [13] explored cross-platform ubiquitous language learning, integrating mobile phones and interactive television (iTV) to bring up learning environment. The system addressed challenges in user interface design and architecture across platforms, focusing on scaffolding difficult language concepts and managing personal learning spaces. Although the system was prototyped and functional on both mobile and iTV interfaces, issues with reading text and on-screen display were noted, particularly on the iTV side. This research was an important step toward achieving truly cross-platform language learning but needed refinement in its display technology. [18] examined the role of natural language processing (NLP) in artificial intelligence, particularly for grammar as a foreign language. By employing a Recurrent Neural Network (RNN) with an attention mechanism, the study generated sentence parse trees to demonstrate that sequence-to-sequence models could excel at syntactic constituency parsing with minimal tuning. The research made a significant contribution to NLP, providing insights into both content and

semantic analysis of large databases. However, the findings were largely theoretical, suggesting that further empirical research is needed to validate these approaches in practical language learning and translation systems. [7] worked on developing a statistical machine translation system for English and Nigerian languages, focusing on English-to-Igbo, English-to-Yorùbá, and Igbo-to-Yorùbá translation. The system was trained on parallel corpora derived from religious texts, achieving BLEU scores of 30.04, 29.01, and 18.72 for the respective language pairs. While the results were promising, the system was limited to the religious domain, and further expansion into other areas of everyday language use was needed to improve its utility and applicability.

[20] created ULearn English which was an open-source platform for learning English vocabulary using incidental acquisition techniques. Adopting a Design Science Research (DSR) approach, the study developed a ubiquitous learning model for vocabulary acquisition. Also, the initial feedback on the system's utility and ease of use was positive, the sample size was small, with only 15 learners participating in the system evaluation. This research highlighted the potential of open, ubiquitous learning systems but required broader testing for more generalizable results. [10] tried to improve English-Arabic translation using the transformer model with multi-head attention. This approach combined a feed-forward network with attention mechanisms, resulting in accuracy of 97.68% and a BLEU score of 99.5. Despite the model's success, challenges remained, particularly in low-data resource languages, handling rare words, and improving domain adaptation. These limitations pointed to ongoing issues in scaling transformer-based models to handle diverse languages effectively. [29] focused on Natural Language Interface (NLI) for generating data flow diagrams (DFDs) using web extraction techniques. By applying NLP techniques to scraped data, the system identified key terms and mapped user queries to the appropriate DFD structures. Empirical testing showed that the system-generated DFDs were accurate and complete. However, the study did not consider additional variables such as time and cost efficiency, leaving room for further refinement in its practical application. [21] conducted an in-depth review of data augmentation (DA) techniques in NLP. The study provided a systematic comparison of various DA methods, underlining their importance in enhancing NLP systems' performance. Although the review was comprehensive but it did not give the insights, instead summarizing existing techniques.

III. METHODOLOGY

1. System Architecture

The system is majorly segmented into three parts: Input Phase (Ubiquitous / Mobile Device – System Application Layer / User Interface), Translation Phase (Machine Translation – Transformer Architecture), Output Phase Ubiquitous / Mobile Device - System Application Layer / User Interface). The transformer contains both the encoder and decoder models. The encoder will take the input - English sentence in vector and matrix form. The decoder will take in that encoded representation and iteratively generates an output that translates to Yoruba sentence and vice versa.

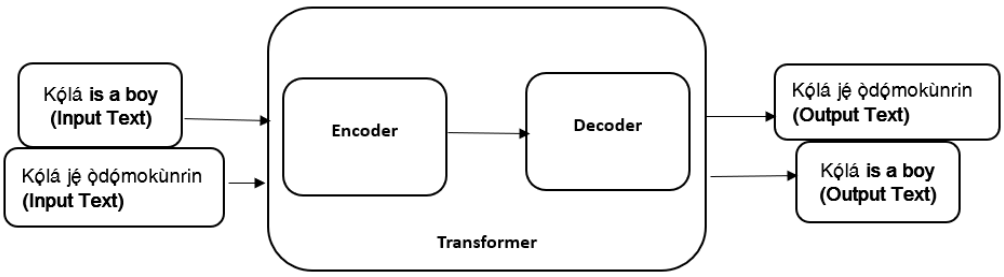

Figure 1: Architecture Overview of ubiquitous bidirectional translation for language learners

The transformer contains the encoder and decoder of the model. In which the encoder takes the input in vector and matrix form. Representation of the input like an English sentence "Kola is a boy", the decoder takes in that encoded representation and iteratively generates result and translates the sentence to "`K-lq j1 =d-mok6nrin`".

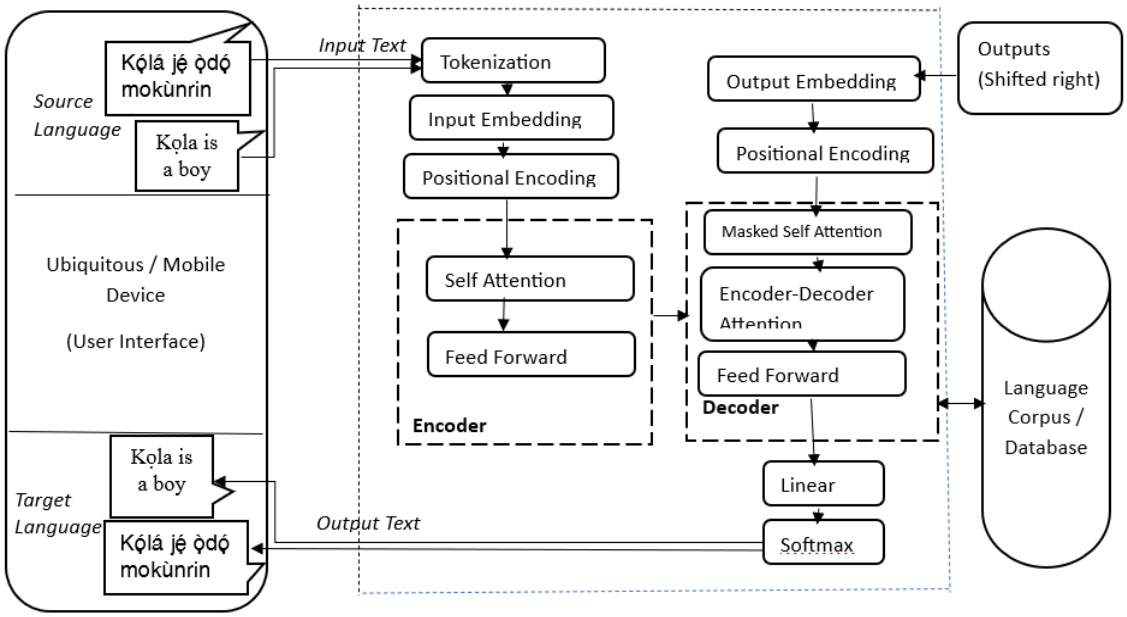

Figure 2: Architecture of ubiquitous bidirectional translation for language learners

2. User Interface

System Application Layer is an ubiquitous/mobile based application used for collecting input data which is source language from the user before it is processed for translation, It also shows the output of the text translated which is target language. Figure 3 depicts a typical application system which is made up of ubiquitous or mobile devices and how it links with other sections.

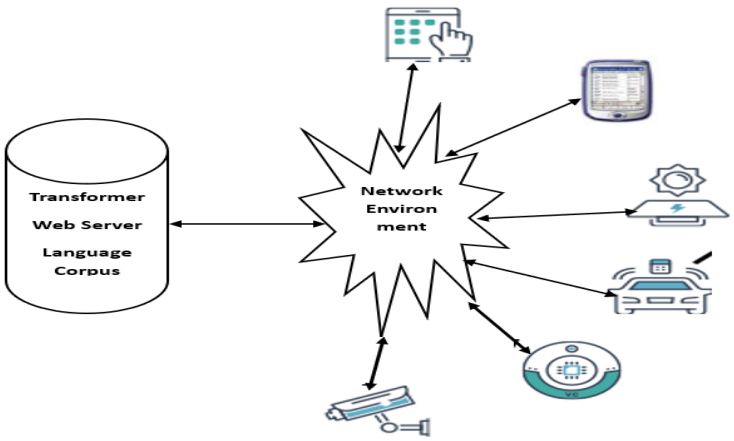

Figure 3: Ubiquitous devices linked with other sections

3. Translation Segment

3.1 Encoder Phase

3.1.1 Lexical Analysis - Scanner

Which allows the breaking or splitting of paragraphs and sentences into individual smaller units or words or tokens that can be more easily assigned meaning with their unique properties for easy identification and translation – Tokenization. Whitespace tokenization is the easiest and frequently used form of tokenization. It separates the text whenever it finds whitespace characters. Unigram algorithm always keeps the base characters so that any word can be tokenized. Building a Tokenizer, there are basically ways of building a tokenizer: careful coding by hand, if the tokenizer is not big, this approach can be followed and using of a scanner generator (Lex).

3.1.2 Tokenization

Tokenization refers to splitting a text into its fragments - usually single words. Breaking Text into meaningful chunks or fragments, these fragments are called tokens. The model uses special tokens such as Start of Sentence [SOS] or [CLS] for the start of the sequence and End of sentence [EOS] or [SEP] for the end of the sequence, this helps to make sense of the input. Text is needed to be broken down into smaller units called tokens. It splits a sentence/phrase into smaller pieces or words, like "Kola" + "is" + "a" + "boy".

3.1.3 Input Embedding:

The encoder begins by converting input tokens - words or sub words - into vectors using embedding layers. These embeddings capture the semantic meaning of the tokens and changed them into numerical vectors. All the encoders receive a list of vectors. In the lowest encoder, this input would consist of word embeddings, whereas in subsequent encoders, it would be the output from the encoder directly beneath it.

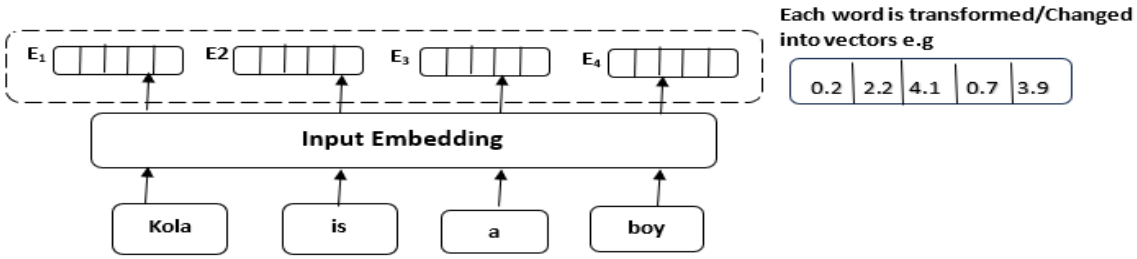

Figure 4: Input Embedding

3.1.4 Positional encoding

The input embeddings make provision for information about the position of individual token in the sequence. This makes the model to understand the position of each word within the sentence, since Transformers do not have a recurrence mechanism like RNNs (Recurrent Neural Network). The combination of various sine and cosine functions to create positional vectors, enabling the use of this positional encoder for sentences of any length. Suppose the input sequence of length L and requires the position of the kth object in the sequence. The positional encoding is described by sine and cosine functions of varying frequencies:

$$P(k, 2i) = sin\left(\frac{k}{n^{2i/d}}\right) \quad (1)$$

$$P(k, 2i+1) = cos\left(\frac{k}{n^{2i/d}}\right) \quad (2)$$

Where k is the position of an object or token in the input sequence or text, d is the dimension of the output embedding space, $P(k, j)$ is the Position function for mapping a position k in the input sequence to index (k, j) of the positional matrix, j can be $2i$ or $2i+1$, n is User-defined scalar number, i is used for mapping to column indices $0 <= i < d/2$ with a single value of i maps to both sine and cosine functions.

From the expression above, odd positions correspond to cosine functions, also, even positions correspond to a sine function. For example, the phrase "Kola is a boy," with n=100 and d=4. The table below illustrates the positional encoding matrix for this phrase. Notably, this positional encoding matrix remains the same for any four-character phrase with n=100 and d=4

Table 1 Position Encoding for the sentence "Kola is a boy"

Sequence	Index of Token K	i = 0	i = 0	i = 1	i = 1
Kola	0	P_{00} = Sin(0) = 0	P_{01} = Cos(0) = 1	P_{02} = Sin(0) = 0	P_{03} = Cos(0) = 1
Is	1	P_{10} = Sin(1/1) = 0.84	P_{11} = Cos(1/1) = 0.54	P_{12} = Sin(1/10) = 0.10	P_{13} = Cos(1/10) = 1.0
A	2	P_{20} = Sin(2/1) = 0.91	P_{21} = Cos(1/1) = -0.42	P_{22} = Sin(2/10) = 0.20	P_{23} = Cos(2/10) = 0.98
boy	3	P_{30} = Sin(3/1) = 0.14	P_{31} = Cos(1/1) = -0.99	P_{32} = Sin(3/10) = 0.30	P_{33} = Cos(3/10) = 0.96

3.1.5 Self Attention

This operates by transforming the input sequence into three vectors: query- Q, key- K and value - V. Query (Q) is a vector that represents a specific word or token looking for other words or tokens to pay attention to. Key (K) is also a vector looked at by another words. Value (V) is the information or meaning of the word. These vectors are derived from linear transformations of the input. The attention mechanism calculates a weighted sum of the values by assessing the similarity between the query and key vectors, thereby improving the model's capacity to comprehend context. The larger size of the dot product is kept from dominating and overshadowing the others by attention score scaling.

To ensure more steady gradients, the score is divided by the square root of the query and key dimensions as shown in equation 3 and 4. This is done to prevent the multiplying numbers from likely explosive effects.

$$Scaling\ of\ Similarity\ score\ = \frac{QK^T}{\sqrt{d_k}} \tag{3}$$

$$Attention(Q,\ K,\ V) = SoftMax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

d_k is the dimension of the key vectors.

Table 2: Self Attention Position

	Kola	Is	a	Boy
Kola	98	27	10	12
is	27	89	31	67
a	10	31	91	54
boy	12	67	54	92

Table 3: QK^T is dot product between Query and Key

	Kola	is	a	Boy
Kola	0.7	0.1	0.1	0.1
Is	0.1	0.6	0.2	0.1
A	0.1	31	0.6	0.1
Boy	0.1	0.3	0.3	0.3

SoftMax of the Scaled Scores is used to obtain the attention weights and probability values between 0 and 1, the scaled score is SoftMax. SoftMax elevates higher scores while lowering lower ones. By utilizing equation 4, this enables the model to be more certain about which text from the sequence has more attention.

Table 4 individual word (Q) in relation to the other words in the sentence (K, V)

Kola	is	a	Boy
Kola	is	a	Boy
Kola	is	a	Boy
Kola	is	a	Boy

And for each word in the sentence the Transformer creates the Q, K, V vector. It represents an individual word (Q) in relation to the other words in the sentence (K, V). As an example for a sentence "Kola is a boy", following Q, K, V vectors may be created. e.g.

* Q1 = "Kola" and K,V = ("is", "a", "boy")

Once the query vector Q1 = "kola" is processed by the encoder side, the Query vector moves to the next word Q2 = "is" which is then provided as input to the Encoder, and so on, in a process called "Shifting Focus"

* Q2 = "is" and K,V = ("kola", "a"," boy")
* Q3 = "a" and K,V = ("kola", "is", "boy")
* Q4 = "boy" and K,V = ("kola", "is", "a")

3.1.6 Feed Forward

After the self-attention mechanism, the output moves to a feed-forward neural network (FFN). Combination of linear layers, with a ReLU (Rectified Linear Unit) activation nestled in between it, acting as a bridge. Once processed, the output embarks on a familiar path: it loops back and merges with the input of the pointwise feed-forward network. many inputs with weights as they enter the layer. The weighted input values

are then summed together. If this sum exceeds a predefined threshold (typically set to zero), the output is generally 1; otherwise, it is -1. A Rectified Linear Unit (ReLU) activation function is applied between these two linear layers to introduce non-linearity. This function is defined as ReLU(x)=max(0,x) and is used to introduce non-linearity into the model, helping it to learn more complex patterns.

$$FFN(x) = \max(0, xW_1+b_1)W_2+b_2 \quad (5)$$

Where W_1 and W_2 are the weight matrices for the first and second linear layers, respectively. b_1 and b_2 are the biases for these layers. The ReLU activation is applied element-wise after the first linear transformation.

$$[y] = [X_1,\ x_2,\ x_3 \ldots\ldots\ldots x_n\ ] \cdot \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ : & & & \\ W_{n1} & W_{n2} & W_{n3} & W_{n4} \end{bmatrix}$$

Figure 5: vectors in matrices form

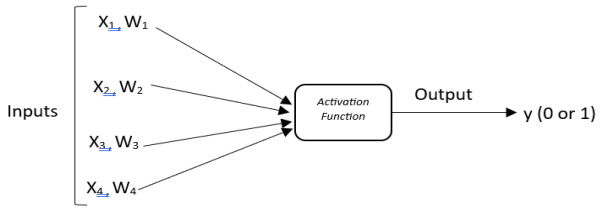

Figure 6: Activation Function with bias vector

Given input x to make a linear transformation to it using a learned weight matrix W (plus a bias vector c) then apply ReLU function on top of it to get hidden layer h, which is used as a linear model with a weight vector w (plus a bias term b) to get final output y. This can actually be written like this:

$$f(x{:}W, c, w, b) = w^T \max(0, W^T x + c) + b \quad (6)$$

The first layer of the FFN transforms this vector into a higher dimensional space, adds a bias, and applies the ReLU activation:

$$x' = \max(0, xW_1+b_1) \quad (7)$$

Assuming non-negative outputs from ReLU for simplicity, the second layer then projects this vector back down to the original dimensional space:

$$\text{FFN output} = x'W_2+b_2 \quad (8)$$

This output is then normalized by a subsequent step and either fed into the next layer of the encoder or used as part of the input to the multi-head attention layer in the decoder.

In a feedforward network, the cost function is a critical component. Minor changes to the weights and biases have minimal impact on the classified data points. As a result, a smooth cost function is often employed to guide adjustments to the weights and biases, helping to enhance the model's performance. One common example of such a cost function is the mean squared error (MSE), defined as follows:

$$C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x) - a\|^2 \quad (9)$$

Where *C(w, b)* = cost function, *w* = the weights gathered in the network, *b* = biases, *n* = number of inputs for training, *a* = output vectors (predicted), *x* = input, *y(x)* = actual output.

3.1.7 Output of the Encoder

The output of the encoder layer is in form of vectors, each representing the input sequence with a rich contextual understanding. This output of the encoder is used as the input for the decoder in a Transformer model. This shows that encoder paves the way for the decoder, leading it to pay attention to the right words in the input when it is time to decode.

The decoder part of the architecture has a structure specifically designed to generate this output by decoding the encoded information step by step. It is important to understand that the decoder functions in an autoregressive manner, initiating its process with a start token. It relies on a sequence of previously generated outputs as input, along with attention-based information from the encoder corresponding to the earlier input.

3.2 Decoder Phase

3.2.1 Output Embeddings:

At the decoder's starting line, the process mirrors of that of the encoder. Here, the input first passes through an embedding layer. These embeddings capture the semantic meaning of the tokens and change them into numerical vectors. All the decoders receive a list of vectors. In the bottom decoder, that would be the word embeddings.

3.2.2 Positional Encoding:

Following the embedding, again just like the encoder, the input passes by the positional encoding layer. This sequence is designed to produce positional embeddings. These positional embeddings are then channeled into the first multi-head attention layer of the decoder, where the attention scores specific to the decoder's input are meticulously computed sentences of any length. Consider an input sequence of length *L*, and the goal is to determine the position of the *kth* element within this sequence. The positional encoding for this element is represented using sine and cosine functions at different frequencies.:

$$P(k, 2i) = sin\left(\frac{k}{n^{2i/d}}\right) \quad (10)$$

$$(k, 2i + 1) = cos\left(\frac{k}{n^{2i/d}}\right) \quad (11)$$

k represents the position of the token in the input sequence, d denote the dimensionality of the output embedding space. The variable j corresponds to either *2i* or *2i+1*, where *i* is an index used to map column indices such that 0 <= i < *d/2*. The positional function *P(k, j)* maps the position k in the input sequence to the index *(k, j)* of the positional encoding matrix, *j* can be *2i* or *2i+1*. Additionally, *n* is a user-defined scalar value, and *i* is utilized to associate a single index with both the sine and cosine functions for encoding positional information.

3.2.3 Masked Self-Attention:

This is same with self-attention in the encoder but with a crucial difference: it prevents positions from attending to subsequent positions, which means that each word in the sequence is not influenced by future tokens. For instance, when the attention scores for the word "is" a is being computed, it's important that "is" does not get a peek at "a", which is a subsequent word in the sequence. This masking makes sure that the particular position prediction is only rely on known outputs at positions before it.

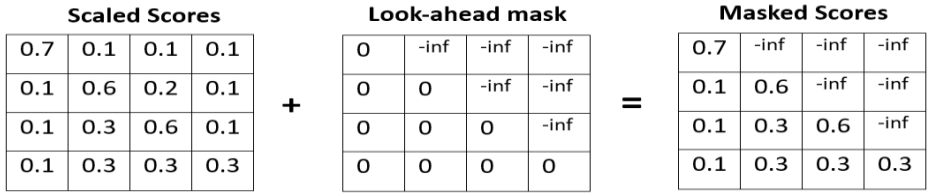

Figure 7: Masked Self-Attention

3.2.4 Encoder-Decoder Attention or Cross Attention

In the attention layer of the decoder, there is a unique interplay between the encoder and decoder's components. Here, the outputs from the encoder take on the roles of both queries(Q) and keys(K), while the results from the first multi-headed attention layer of the decoder serve as values (V). This setup effectively aligns the encoder's input with the decoder's, empowering the decoder to identify and emphasize the most relevant parts of the encoder's input. Following this, the output from this second layer of attention is then refined through a pointwise feedforward layer, enhancing the processing further.

3.2.5 Feed-Forward Neural Network

This just like the encoder, each decoder layer includes a fully connected feed-forward network, applied to each position individually and identically.

3.3 Linear Classifier and SoftMax for Generating Output Probabilities

The journey of data through the transformer model culminates in its passage through a final linear layer, which functions as a classifier. The resulting vectors of the linear layer are called logits vectors.

3.3.1 SoftMax layer

The SoftMax function is calculated using the following equation:

$$F(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}} \quad (12)$$

Where x_i is input vector, e^{x_i} is standard exponential vector for input vector, k is number of classes in multiclass classifier, e^{x_j} is standard exponential vector for output vector, $i = 0, 1, 2, \ldots, k$, $x_i = x_1, x_2, \ldots, x_k$ is the input value and x_j is the output of the values in the input. For any input, the outputs must all be positive and they must sum to unity.

Those logits vectors are then transferred to the SoftMax layer that changes the values of each index into probabilities. The size of this classifier relates to the total number of classes involved (number of words

contained in the vocabulary). For instance, in a scenario with 5000 distinct classes representing 5000 different words, the classifier's output will be an array with 5000 elements. This output is then introduced to a SoftMax layer, which transforms it into a range of probability scores, each lying between 0 and 1. The highest of these probability scores is key, its corresponding index directly points to the word that the model predicts as the next in the sequence. Each sub-layer (masked self-attention, encoder-decoder attention, feed-forward network) is followed by a normalization step, and each also includes a residual connection around it.

3.4 Translation of output scores or vectors into text

The output from the decoder stack is a vector with the same dimensions as the word embeddings provided at the input stage. During inference, this vector must be converted into a word at each step. This task is handled by two layers: the linear layer and the SoftMax layer. The linear layer, which is a fully linked to neural network, maps the output vector to a high-dimensional space where each dimension represents a word in the model's corpus. The values within this space shows the likelihood of each word being the correct output, similar to the approach used in bag-of-words models. The corpus consists of all the words the model encountered during training.

The logits produced by the linear layer are then passed to the SoftMax layer, which converts these values into a probability distribution. Each index in the distribution represents the probability of a specific word being the next output. The Argmax function is applied to select the index with the highest probability, and the word associated with this index is retrieved from the vocabulary.

Then the sequence of translated text which can be in form of English to Yoruba or vice versa since the application is Bidirectional translation will be displayed on any mobile or ubiquitous device that the application has been installed for Language translation and learning purposes.

3.5 Training and Testing of data

Data will be collected from religious book, Yoruba - English Dictionary and other necessary documents to build the corpus or database. The dataset will be split into two subsets: 80% for training and 20% for testing. The training process will utilize the transformer model, a powerful neural network architecture specifically designed for handling sequence-to-sequence tasks in natural language processing. Unlike traditional architectures, the transformer relies on self-attention mechanisms, which allow the model to capture dependencies between input and output tokens without the need for recurrent or convolutional layers. By concentrating on every component of a sequence at once rather of processing each token one after the other, the self-attention technique aids the model in understanding the relationships within the data. The multi-head attention mechanism is one of the transformer model's primary characteristics. With the help of this method, the model can simultaneously focus on distinct portions of the input sequences. The multi-head attention method greatly improves the efficiency of the model by processing this data in parallel, particularly on hardware specifically designed for parallel calculations, like Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs). The transformer model is incredibly efficient for extensive natural language tasks because of its design, which not only improves performance, but also shortens the training period.

IV. CONCLUSION

The research describes improvement on language translation by integrating the Transformer model into the used of ubiquitous bidirectional translation application for learners of Yoruba and English languages. The attention based mechanism of the Transformer is specially suited to managing the challenges posed by the syntactic and sematic structure of English language and Yoruba which is a tonal language. Through integration of transformer architecture helps language translation and learning. User can interact with both languages in real-time, anywhere, anytime on any of the ubiquitous devices and receive results that will helps the development of language learners.

REFERENCES

[1]. Abimbola R. I. and Olufemi D. N. (2017). Development of a Yorúbà Text-to-Speech System Using Festival, Innovative Systems Design and Engineering www.iiste.org, ISSN 2222-1727 (Paper) ISSN 2222-2871, Vol.8, No.5, 2017

[2]. Afolabi A. and Wahab A. (2013) Implementation of Yoruba Text-To-Speech E-Learning System. International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 11, IJERT ISSN: 2278-0181

[3]. Akinlabi A. (2004) The sound system of Yoruba, Yoruba language Phonology. Understanding Yoruba life and culture, Proceedings of the 4th World Congress of African Linguistics, New Brunswick 2004.

[4]. Akinwale, O., Adetunmbi, A. O., Obe, O., and Adesuyi, A. (2015). Web-based english to yoruba machine translation. International Journal of Language and Linguistics, 3(3):154-159.

[5]. Akinwonmi E. A. (2021). Development of a Prosodic Read Speech Syllabic Corpus of the Yoruba Language. Communications on Applied Electronics (CAE) – ISSN: 2394-4714 Foundation of Computer Science FCS, New York, USA Volume 7 – No. 36, June 2021 – www.caeaccess.org

[6]. Ally, M., & Prieto-Blázquez, J. (2014). What is the future of mobile learning in education? *The International Journal of Educational Technology in Higher Education*, *11*(1), 142-151. https://doi.org/10.7238/rusc.v11i1.2033.

[7]. Ayogu I. I. , Adetunmbi A. O. and Ojokoh B. A. (2018) Developing Statistical Machine Translation System for English and Nigerian Languages. Asian Journal of Research in Computer Science 1(4): 1-8, 2018; Article no. AJRCOS.44217.

[8]. Bahdanau D., Cho K, and Bengio Y (2024). Neural Machine Translation by Jointly Learning to Align and Translate. Computer Science Computation and Language. arXiv preprint arXiv:1409.0473, 2014.

[9]. Biyi F., Jillian C. and Mi Z. (2018). DeepASL: Enabling Ubiquitous and Non-IntrusiveWord and Sentence-Level Sign Language Translation. SenSys '17, November 6–8, 2017, Delft, Netherlands (13), https://doi.org/10.1145/3131672.3131693.

[10]. Donial G. , Marco A., Salud M. and Mostafa A. (2023) Case Study of Improving English-Arabic Translation Using the Transformer Model. International Journal of Intelligent Computing and Information Sciences 23(2):105-115, DOI: 10.21608/ijicis.2023.210435.1270

[11]. Eludiora S. I. and Agbeyangi A. O. (2015). Development of English to Yorùbá Machine Translation System for Yorùbá Verbs' Tone Changing. International Journal of Computer Applications (0975 – 8887) Volume 129 – No.10.

[12]. Enriquez J. J. (2018). Natural language processing in artificial intelligence (NLP AI) and natural language processing algorithms relating to grammar as a foreign language. Jurnal Pendidikan Profesi Guru Indonesia. https://www.researchgate.net/publication/328268746

[13]. Fallahkhair S., Pemberton L. and Griffiths R. (2017) Development of a cross-platform ubiquitous language learning service via mobile phone and interactive television. The Authors. Journal compilation, Blackwell Publishing Ltd Journal of Computer Assisted Learning, 23, 312–325 doi: 10.1111/j.1365-2729.2007.00236.x.

[14]. Finch A. and Sumita E. (2010). Transliteration using a Phrase-based Statistical Machine Translation System to Re-score the Output of a Joint Multigram Model. Proceedings of the 2010 Named Entities Workshop, ACL 2010, pages 48–52, Uppsala, Sweden, 2010 Association for Computational Linguistics.

[15]. Guptaa C., Jainb A. and Joshic N. (2018) Fuzzy Logic in Natural Language Processing – A Closer View. International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Procedia Computer Science 132 (2018) 1375–1384

[16]. Humaid A. And Veton K. (2020) Multilingual speech-to-speech translation System in Mobile Offline Environment. Journal of Engineering Research and Application www.ijera.com ISSN : 2248-9622, Vol. 10, Issue 4, (Series - III), pp. 45-50

[17]. Isewon I., Oyelade J. and Oladipupo O. (2014) Design and Implementation of Text to Speech Conversion for Visually Impaired People. International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 7– No. 2, April 2014 – www.ijais.org

[18]. Krasimir A., Bj¨orn B. and Aarne R. (2014). Speech-Enabled Hybrid Multilingual Translation for Mobile Devices. Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 41–44, Gothenburg, Sweden, Association for Computational Linguistics.

[19]. Laurent B., Etienne B., Alexey K. and Tanja S. (2017) Automatic Speech Recognition for Under-resourced Languages: A survey. Published by Elsevier in Speech Communication Vol. 56, 85-100 https://doi.org/10.1016/j.specom.2013.07.008

[20]. Letícia G. S., Eduardo G. A., Rosemary F., Jorge L. V. B., Luis A. S. and Valderi R. Q. L. (2021). ULearnEnglish: An Open Ubiquitous System for Assisting inLearning English Vocabulary. Electronics 2021, 10, 1692. https://doi.org/10.3390/electronics10141692.

[21]. Lucas F. A. O. P., Taynan M. F. and Anna H. R. C. (2023) Augmentation Techniques in Natural Language Processing. Applied Soft Computing 132 (2023) 109803. www.elsevier.com/locate/asoc.

[22]. Muthalib A., Anas A., Mohammad S., and Juhriyansyah D. (2011). Making Learning Ubiquitous with Mobile Translator Using Optical Character Recognition (OCR). ICACSIS 2011, ISBN: 978-979-1421-11-9

[23]. Nyetanyane J. and Masinde M. (2018). UmobiTalk: Ubiquitous Mobile Speech Based Translator for Sesotho Language. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering pp. 93–106.

[24]. Omar A. (2017). Using Objective Words in the Reviews to Improve the Colloquial Arabic Sentiment Analysis. International Journal on Natural Language Computing (IJNLC) Vol. 6, No.3, June 2017

[25]. Ott M., Edunov S., Grangier D. and Michael A. (2018). Scaling Neural Machine Translation. Proceedings of the Third Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 1–9 Belgium, Brussels, October 31 - November 1, 2018. Association for Computational Linguistics, https://doi.org/10.18653/v1/W18-64001.

[26]. Parhizkar B., Oteng K., One N., Arash Habibi L. and Zahra M. (2014) Ubiquitous Mobile Real Time Visual Translator Using Augmented Reality for Bahasa Language. International Journal of Information and Education Technology, Vol. 3, No. 2. DOI: 10.7763/IJIET.2013.V3.248.

[27]. Sandeep R. W., Prakash R. D. and Patil S. H. (2012). English-to-Sanskrit Statistical Machine Translation with Ubiquitous Application. International Journal of Computer Applications (0975 – 8887) Volume 51– No.1

[28]. Sandeep R. W., Prakash R. D. and Patil S. H. (2013). Language Learning and Translation with Ubiquitous Application Through Statistical Machine Translation Approach. International Journal of Advances in Engineering & Technology Vol. 4, Issue 2, pp. 474-481

[29]. Sehrish M. C., Saman T. and Ivan M. P. (2023). A natural language interface for automatic generation of data flow diagram using web extraction techniques. Journal of King Saud University – Computer and Information Sciences 35 (2023) 626–640.

[30]. Théophile K. D and Charbel B. (2014). A Text to Speech System for Fon Language Using Multisyn Algorithm. Procedia Computer Science Volume 35, 2014, Pages 447-455

[31]. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Llion J., Aidan N. G., Lukasz K. and Illia P. (2017) Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA

[32]. Xue W., Wei Z. and Xinhui Y. (2017) Construction of Course Ubiquitous Learning Based on Network. EURASIA Journal of Mathematics Science and Technology Education, ISSN: 1305-8223 (online) 1305-8215, 2017 13(7):3315-3323, DOI 10.12973/eurasia.2017.00728a.